X-FENO ASSEMBLY PLATFORM (MODIFIED 3D PRINTER)

Carlos Omeñaca Prado

January 2026

Abstract

This document presents the design, assembly, and operation of the X-FENO Assembly Platform, a low-cost, open, and reproducible system for deterministic fabrication of twisted two-dimensional (2D) material heterostructures (graphene, hBN, and related van der Waals stacks). The platform is built by modifying a standard Cartesian desktop 3D printer, replacing the extrusion toolhead with a rigid scientific tool stack integrating a piezoelectric XYZ micro-manipulator, a motorized rotation stage for twist-angle control, and a PDMS/PPC-based pick-up and transfer tool.

The system enables micron-scale positioning, sub-degree rotational alignment, controlled thermal contact, and vacuum-assisted substrate handling, following the same physical principles used in professional van der Waals heterostructure laboratories. All mechanical components, electronics, materials, and procedures are explicitly specified to allow independent replication without proprietary parts or calibration-dependent assumptions.

The total cost of the complete platform is in the range of €350–900, representing an order-of-magnitude reduction (approximately 90–95%) compared to commercial graphene transfer and stacking systems, while preserving full functional capability for research, educational, and exploratory device fabrication. Optional extensions for automation, computer vision–assisted alignment, and software-controlled operation are included.

This work is intended as a practical engineering disclosure: a fully constructible machine, a validated operational workflow, and a foundation for accessible experimental research on twisted 2D material devices.

Contents

1 ASCII OVERVIEW OF THE X-FENO ASSEMBLY MACHINE **5**
- 1.1 LEGEND OF COMPONENTS (brief) . . . 6
- 1.2 Purpose of the schematic . . . 6

2 FULL BILL OF MATERIALS (BOM) **7**
- 2.1 A. BASE MACHINE (3D PRINTER FRAME) . . . 7
- 2.2 B. ENCLOSURE & ENVIRONMENT CONTROL . . . 7
- 2.3 C. TOOLHEAD BLOCK (REPLACES EXTRUDER) . . . 8
- 2.4 D. MICRO-MANIPULATOR (SUB-MICRON XYZ STAGE) . . . 8
- 2.5 E. ROTATION MODULE (TWIST ANGLE CONTROL) . . . 8
- 2.6 F. PDMS/PPC TRANSFER TOOL . . . 8
- 2.7 G. VACUUM CHUCK + HOT STAGE . . . 9
- 2.8 H. MICROSCOPY + ALIGNMENT . . . 9
- 2.9 I. CVD LOCALIZED DEPOSITION (OPTIONAL) . . . 9
- 2.10 J. CABLES, ELECTRONICS & SAFETY . . . 9
- 2.11 K. MATERIALS FOR X-FENO FABRICATION . . . 10
- 2.12 L. TOOLS REQUIRED FOR ASSEMBLY . . . 10

3 TOTAL COST RANGE **10**

4 ASSEMBLY OF THE MACHINE **10**
- 4.1 Prepare the donor 3D printer . . . 10
- 4.2 Install the Toolhead Block . . . 15
- 4.3 Mount the Micro-Manipulator (Piezo XYZ Stage) . . . 18
- 4.4 Install the Rotation Module . . . 19
- 4.5 Attach the PDMS/PPC Stamp Tool . . . 22
- 4.6 Build the Vacuum Chuck + Hot Stage . . . 25
- 4.7 Enclosure Assembly . . . 27
- 4.8 Mount the Microscope and Alignment Lighting . . . 27
- 4.9 Electrical Integration (High-Level) . . . 28
- 4.10 Initial Calibration . . . 29

5 OPERATING PROCEDURE FOR FABRICATING X-FENOS **29**
- 5.1 Overview of the Full X-Feno Workflow . . . 29
- 5.2 Phase 1 — Substrate Preparation . . . 30
- 5.3 Phase 2 — PDMS/PPC Stamp Preparation . . . 30
- 5.4 Phase 3 — Pick-Up of the First 2D Layer . . . 31
- 5.5 Phase 4 — Twist-Aligned Stacking (Core X-Feno Step) . . . 33
- 5.6 Phase 5 — Final Release, Cooling, and Cleaning . . . 34
- 5.7 Phase 6 — X-Feno Final Characterization . . . 35
- 5.8 Summary Workflow . . . 35

6 SAFETY, COMMON ERRORS, TROUBLESHOO-TING **35**
- 6.1 Safety Guidelines . . . 35
- 6.2 Common Errors and How to Fix Them . . . 36
- 6.3 Troubleshooting Flowchart . . . 38
- 6.4 Quick Checklist Before Each Fabrication Run . . . 39

7 AUTOMATION: generating G-code and microcontroller scripts for the X-feno machine **40**
7.1 Control architecture: what talks to what . . . 40
7.2 Coordinate system and calibration . . . 40
7.3 Custom G-code conventions for the X-feno machine . . . 41
7.4 Example G-code: basic building blocks . . . 41
7.5 Host-side automation: Python script . . . 42
7.6 Simple state machine on the microcontroller . . . 42

8 HOW TO ADD BASIC COMPUTER VISION (DETECTING FLAKE POSITION AND ALIGNMENT AUTOMATICALLY) **43**
8.1 System architecture . . . 43
8.2 Calibration: pixels to millimeters . . . 44
8.3 Basic flake detection pipeline . . . 44
8.4 Selecting the target flake automatically . . . 45
8.5 Convert pixel coordinates to machine coordinates . . . 45
8.6 Auto-alignment of rotation angle . . . 46
8.7 Closed-loop operation . . . 46
8.8 Example full Python script . . . 46
8.9 Optional improvements . . . 47

9 ADDING A SIMPLE USER INTERFACE (GUI) FOR AUTOMATED X-FENO FABRICATION **47**
9.1 Overall goals of the GUI . . . 47
9.2 ASCII mockup of the GUI layout . . . 48
9.3 Core GUI loop (Tkinter + OpenCV) . . . 48
9.4 Adding flake detection overlay . . . 49
9.5 Buttons and actions . . . 49
9.6 Manual flake selection with mouse click . . . 50
9.7 G-code output log . . . 50
9.8 Reading machine coordinates (optional) . . . 50
9.9 Integrating auto-alignment . . . 50
9.10 Full automated fabrication . . . 50
9.11 Emergency STOP implementation . . . 51
9.12 Future expansions . . . 51

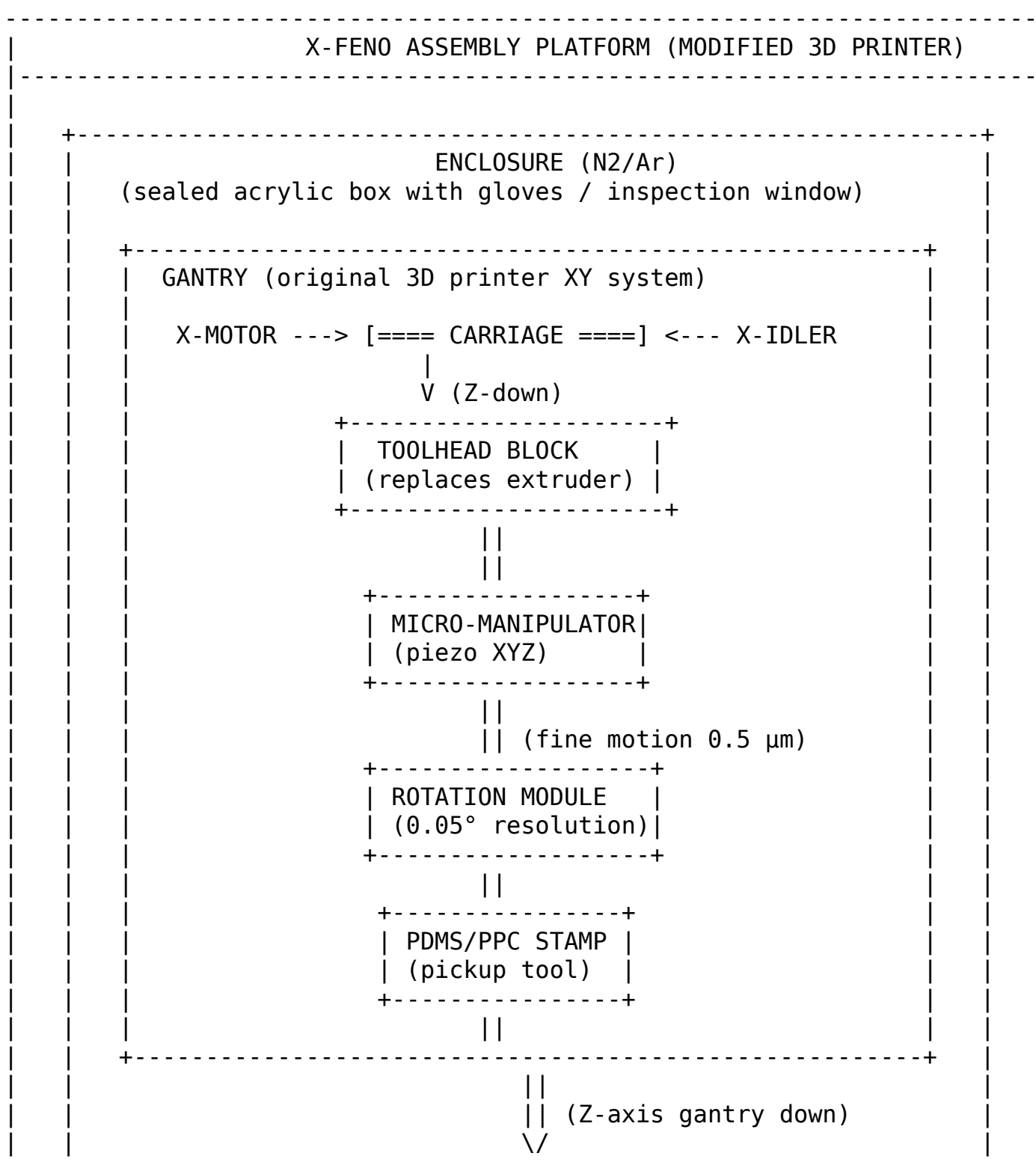
10 FULL SOFTWARE PACKAGE **51**
10.1 Project tree . . . 51
10.2 main.py (central entry point) . . . 53
10.3 settings.yaml (configuration file) . . . 53
10.4 vision/camera_stream.py . . . 53
10.5 vision/flake_detection.py . . . 54
10.6 vision/flake_alignment.py . . . 54
10.7 machine/gcode_sender.py . . . 54
10.8 automation/pickup_routine.py . . . 55
10.9 automation/drop_routine.py . . . 55
10.10automation/full_cycle.py . . . 55
10.11gui/gui_main.py . . . 56
10.12utils/math_utils.py . . . 57

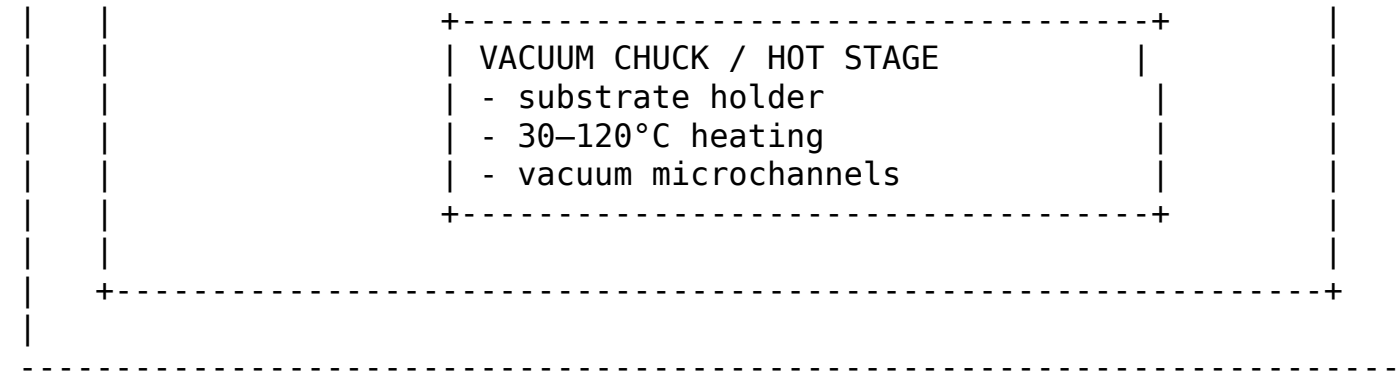
11 Reference **57**

1 ASCII OVERVIEW OF THE X-FENO ASSEMBLY MACHINE

Below is a **large ASCII schematic** showing the entire machine: printer frame, XYZ gantry, toolhead, micro-manipulator, rotation module, PDMS stamp, hot stage, vacuum chuck, and enclosure.

Use a monospaced font.





1.1 LEGEND OF COMPONENTS (brief)

TOOLHEAD BLOCK : replaces extruder; mounts scientific modules
MICRO-MANIPULATOR : piezo XYZ stage for sub-micron adjustments
ROTATION MODULE : rotates 2D materials for twist-angle control
PDMS/PPC STAMP : picks up & releases graphene/hBN
VACUUM CHUCK : holds wafers/substrates with negative pressure
HOT STAGE : releases PPC and anneals stack
ENCLOSURE : keeps N_2/Ar clean atmosphere
GANTRY : original 3D printer XY system

Figure 1: Donor 3D printer (baseline machine)

1.2 Purpose of the schematic

This **macro layout** defines:

- where every subsystem physically sits,
- how they interact during assembly of X-fenos,
- the spatial logic of the machine,
- and the relationship between precision modules (piezo + rotation) and large-scale motion (XY gantry of printer).

2 FULL BILL OF MATERIALS (BOM)

X-FENO ASSEMBLY PLATFORM (Modified 3D Printer)
Version 1.0 — Educational / Research Grade

2.1 A. BASE MACHINE (3D PRINTER FRAME)

These parts come from the donor 3D printer.

```
A1 – 3D Printer Frame (Creality Ender 3 Pro or similar)
A2 – Original X-axis motor + belt
A3 – Original Y-axis motor + belt
A4 – Original Z-axis motor + lead screw
A5 – Original 24V power supply
A6 – Original control board (Marlin-compatible)
A7 – Aluminum bed carriage (Y-axis)
A8 – Linear rails or V-wheels (factory)
A9 – Aluminum extrusion frame
```

Cost: €150–250 (if purchased new)

2.2 B. ENCLOSURE & ENVIRONMENT CONTROL

This ensures a clean atmosphere for 2D-material handling.

```
B1 – Acrylic enclosure panels (3–4 mm thick)
B2 – Aluminum corner brackets
B3 – Silicone seals for airtight closure
B4 – Mini nitrogen/argon inlet valve
B5 – Hygrometer (digital)
B6 – LED illumination strips (white + UV)
B7 – Glove ports or removable front panel
```

Cost: €60–120

2.3 C. TOOLHEAD BLOCK (REPLACES EXTRUDER)

A modular block that mounts all precision tools.

```
C1 – CNC-machined aluminum plate (50×50 mm)
C2 – Tool mounting rail (M3 screw pattern)
C3 – Cable chain (mini, 10 mm)
C4 – 3D-printed adapters (PLA or ABS)
```

Cost: €15–30

2.4 D. MICRO-MANIPULATOR (SUB-MICRON XYZ STAGE)

The core precision system.

```
D1 – Piezo XYZ microstage (resolution < 0.5 µm)
      Example: Thorlabs AE0203D08 (or clone)
D2 – Piezo driver/controller
D3 – Flexible PCB ribbon cables
D4 – Mounting bracket to Toolhead Block
```

Cost: €150–300 (depending on model)

2.5 E. ROTATION MODULE (TWIST ANGLE CONTROL)

Allows graphene layers to be rotated with 0.05° resolution.

```
E1 – Stepper motor (NEMA 8 or NEMA 11)
E2 – Optical rotary encoder (AMS AS5048A or equivalent)
E3 – Small bearing/turntable platform (Ø 20–25 mm)
E4 – Gear reduction ring (3:1 or 5:1 optional)
E5 – Driver board (A4988 or TMC2208)
```

Cost: €30–60

2.6 F. PDMS/PPC TRANSFER TOOL

The pick-up and release tool for 2D materials.

```
F1 – PDMS stamp (custom molded, 2–3 mm thick)
F2 – PPC film (polypropylene carbonate)
F3 – Glass slider mount
F4 – Micro-heater (30–120°C)
F5 – Thermistor + micro temperature controller
```

Cost: €20–40

2.7 G. VACUUM CHUCK + HOT STAGE

Holds substrate flat during transfers.

```
G1 – Machined aluminum chuck plate (80×80 mm)
G2 – Micro vacuum channels (0.5–1.0 mm)
G3 – Vacuum pump (small diaphragm pump)
G4 – Heating pad (120°C max)
G5 – Temperature controller (PID)
G6 – K-type thermocouple
```

Cost: €50–90

2.8 H. MICROSCOPY + ALIGNMENT

For aligning flakes and layers.

```
H1 – USB microscope (1080p or 4K)
H2 – Adjustable arm mount
H3 – Coaxial LED illumination ring
```

Cost: €30–70

2.9 I. CVD LOCALIZED DEPOSITION (OPTIONAL)

For advanced experiments.

```
I1 – Micro-CVD nozzle (3–5 mm aperture)
I2 – Mini resistive microheater (900°C max)
I3 – Low-flow methane inlet
I4 – Stainless steel tubing
```

Cost: €70–150 (optional)

2.10 J. CABLES, ELECTRONICS & SAFETY

General electrical needs.

```
J1 – JST connectors, Dupont cables
J2 – Heat shrink tubing
J3 – Cable sleeve
J4 – DC buck converter 24V→5V and 24V→12V
J5 – On/off rocker switch
J6 – Cooling fan 40 mm
```

Cost: €15–30

2.11 K. MATERIALS FOR X-FENO FABRICATION

Consumables and substrates.

```
K1 – Graphene monolayer on Cu (CVD)
K2 – h-BN flakes (exfoliated or CVD)
K3 – SiO2/Si wafers
K4 – Copper foil (if doing own CVD)
K5 – IPA, acetone, anisole
K6 – Tweezers (anti-static, fine tip)
K7 – Nitrile gloves
```

Cost: €40–120 depending on quality

2.12 L. TOOLS REQUIRED FOR ASSEMBLY

Simple tools to build the machine.

```
L1 – Screwdrivers (PH0, PH1)
L2 – Hex keys (M3, M4)
L3 – Soldering iron + solder
L4 – Wire cutters
L5 – Heat gun
L6 – Small CNC or Dremel (optional)
```

Cost: €0–50 (most labs already have)

3 TOTAL COST RANGE

```
Minimum setup: ~ €350
Full-feature setup: €650–900
Professional-grade: €1200+
```

4 ASSEMBLY OF THE MACHINE

(Modified 3D printer → X-feno assembly platform)

4.1 Prepare the donor 3D printer

1. **Unplug and clear workspace**
   - Disconnect the printer from mains.
   - Remove filament, spool holder, and any accessories.
2. **Remove the extruder toolhead**
   - Unscrew the hotend, heatsink, fan shroud, and filament drive.

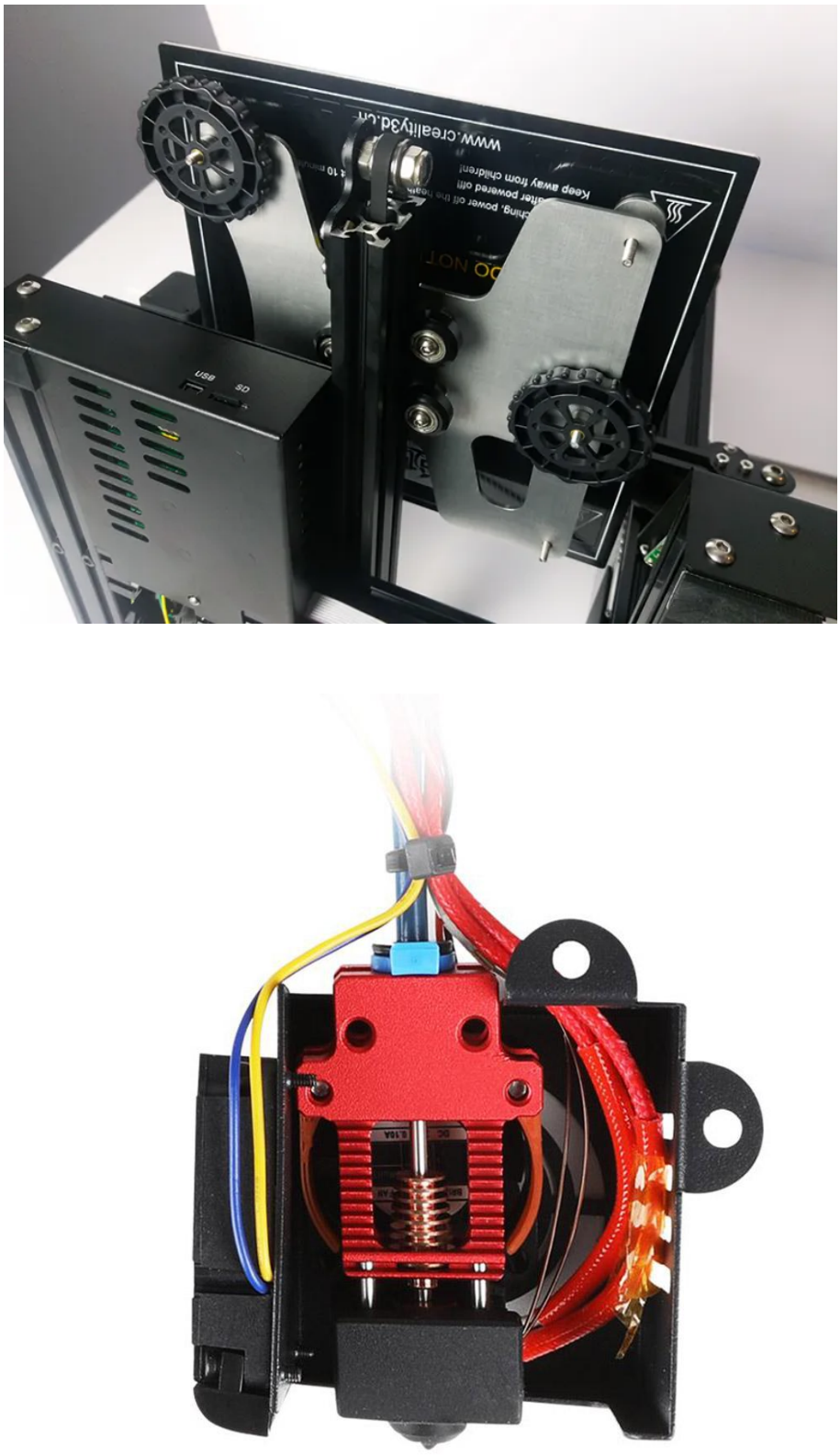


Figure 2: Original extruder toolhead (to be removed)

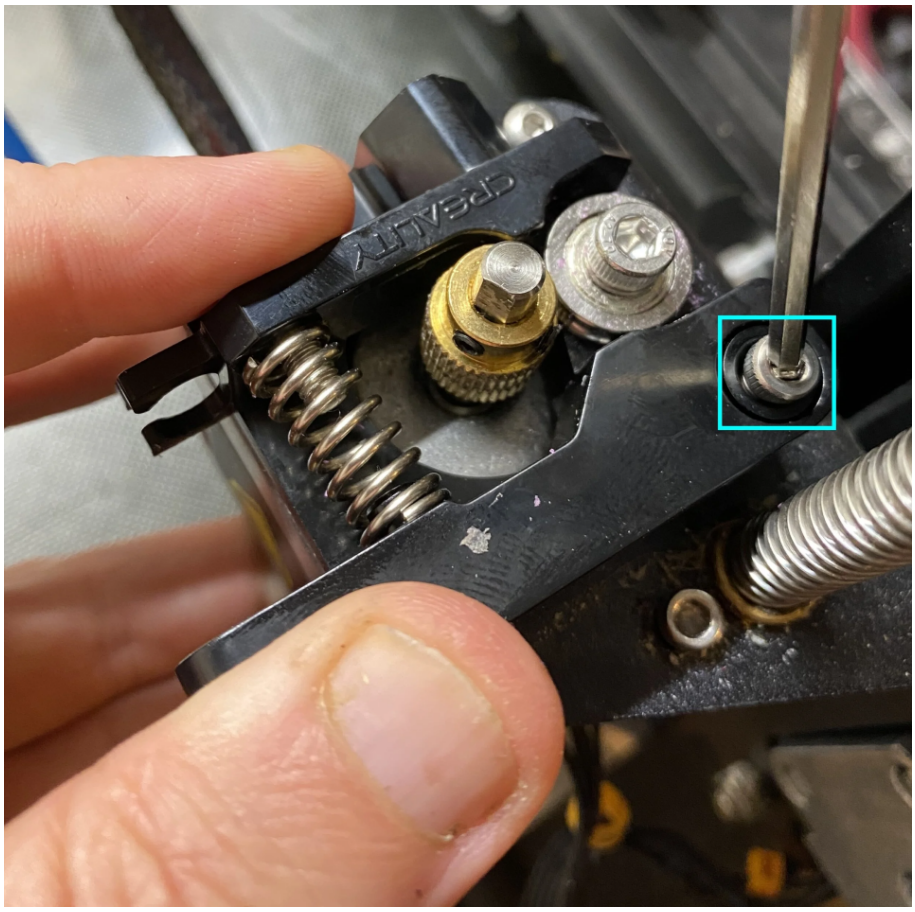


Figure 3: Filament drive mechanism (detail, to be discarded)

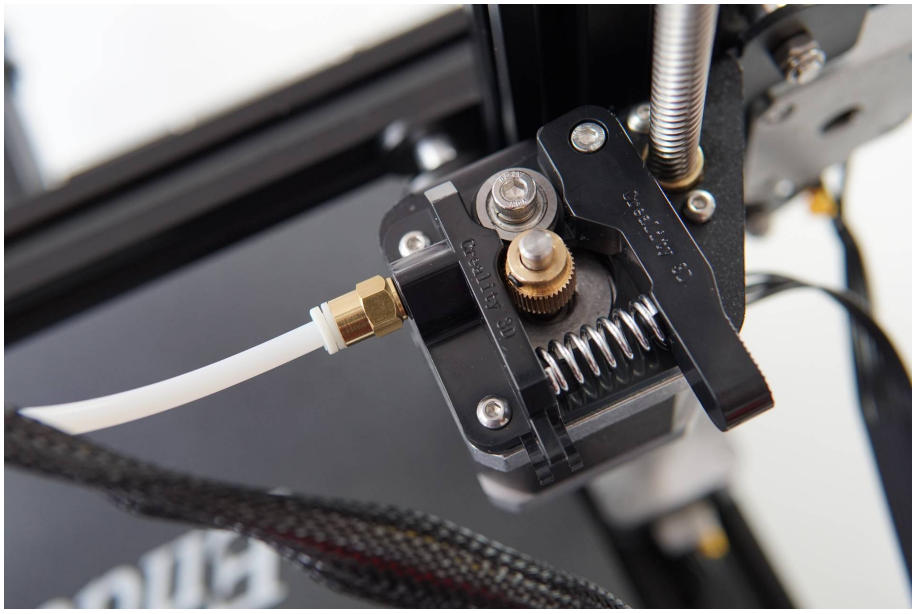


Figure 4: Extruder tensioner and spring assembly (detail)

- Keep:
  - X-carriage plate,

– X-belt and stepper,

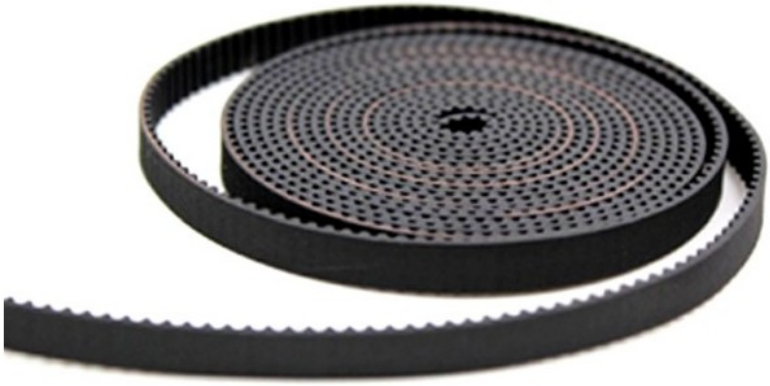
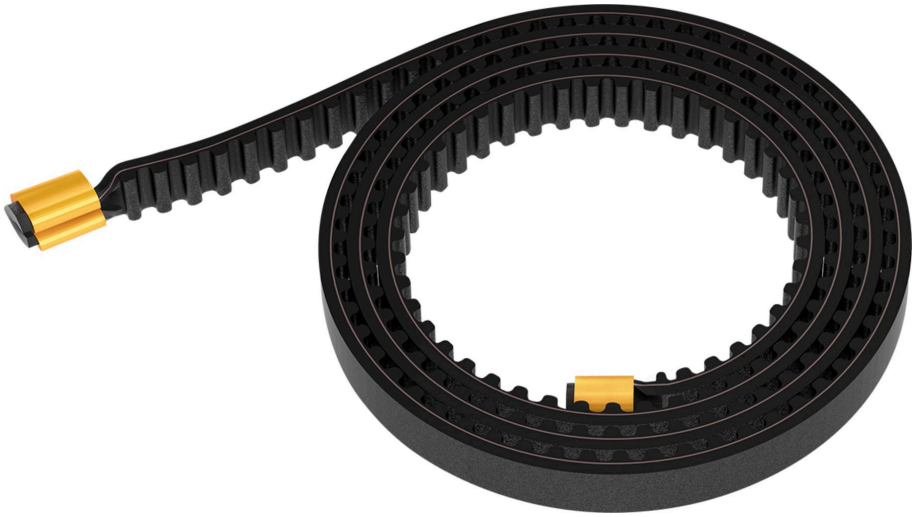


Figure 5: GT2 timing belt 6mm 3D printer

– endstops and wiring.



- You must end with a **bare X-carriage plate**.

Figure 6: stepper motor 42-40

3. **Level the mechanical axes**
    - Manually move X, Y, Z to ensure:
        - No rubbing,
        - No hard friction points,
        - Bed moves smoothly along Y.

Figure 7: Ender 3 X-axis carriage plate

4. **Check firmware**
    - Ensure the printer moves in **absolute coordinates** and can be jogged:
        - X, Y in 0.1 / 1 / 10 mm steps.
        - Z in 0.1 / 1 mm steps.

4.2 Install the Toolhead Block

Goal: replace the extruder by a **rigid scientific tool platform**.

1. **Drill/mark X-carriage if needed**
    - If the original plate doesn't have a flat M3 pattern:

  - Add 4× M3 threaded holes in a 30×30 mm square.
- This is where the Toolhead Block (C1) will fix.

2. **Mount the Toolhead Block**

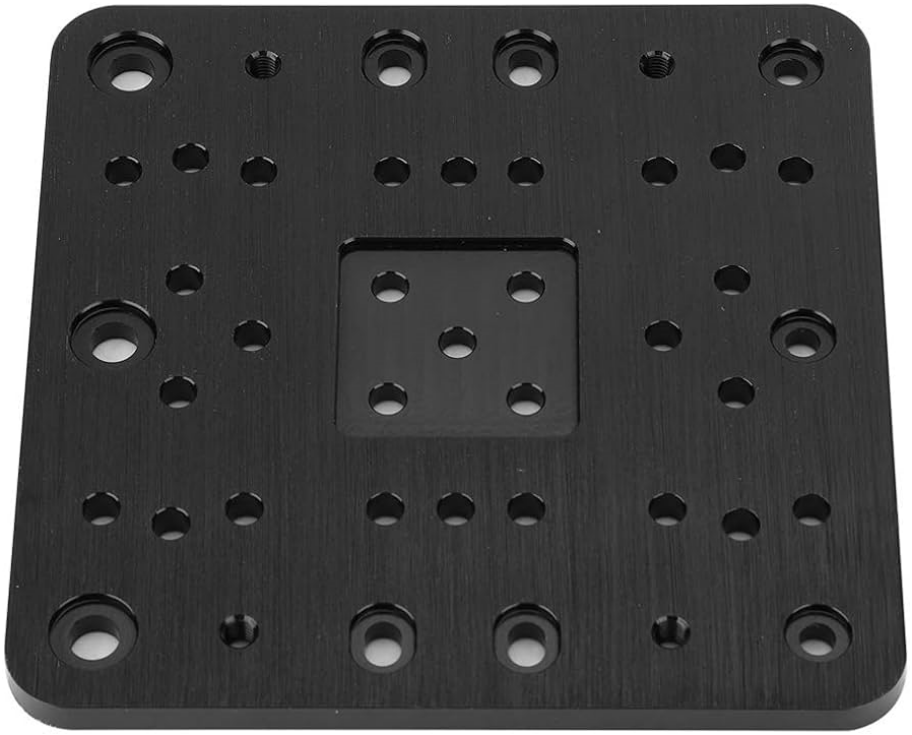


Figure 8: NC aluminum plate 50x50 mm M3

- Fix the aluminum plate (C1) to the X-carriage with 4× M3 screws.
- Orientation:
  - The block should be centered in X,
  - And its front face parallel to the front of the printer.

3. **Add the tool mounting rail**

Figure 9: M3 aluminum mounting rail

- Attach C2 (tool mounting rail) to the front of C1 using M3 screws.
- This rail will carry:
  - Micro-manipulator,
  - Rotation module,
  - Sensor brackets.

ASCII sketch (front view):

```
     (top)
      ^
      |
  +--------------------+   <- X-carriage
  |   C1: TOOLHEAD     |
  |   ALU BLOCK        |
  +--------------------+
         |      |
    M3 screws   |
                +---- C2: tool rail
```

4.3 Mount the Micro-Manipulator (Piezo XYZ Stage)

Goal: fine positioning under the gantry.

1. **Bracket for piezo stage**
    - Use a small L-bracket or 3D-printed part to connect:
        - C2 rail → D1 piezo stage.
    - Fix with M3 screws:
        - Ensure **no play** and **no tilt**.
2. **Attach the piezo stage**

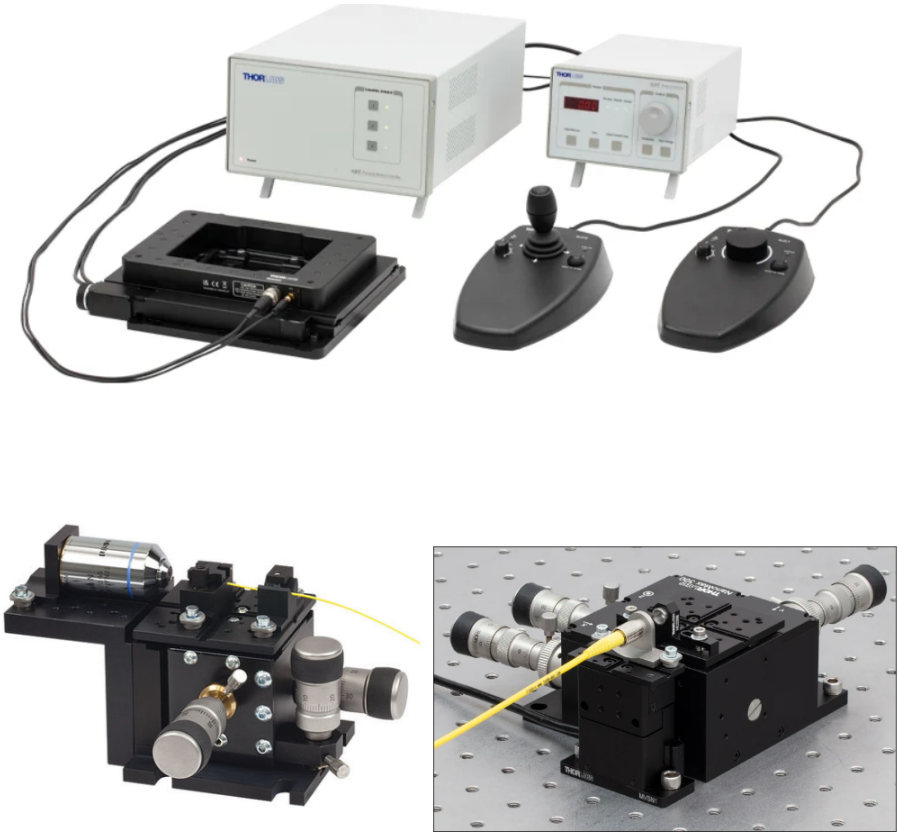


Figure 10: piezo XYZ stage Thorlabs

    - Mount D1 so that its **top surface faces downwards** (towards the substrate).
    - The motion axes:
        - X, Y should be roughly parallel to printer's X, Y.
        - Z should be parallel to printer's Z.
3. **Cable routing**
    - Connect D1 to D2 (piezo driver) with ribbon cables (D3).
    - Route cables through a **mini cable chain** to avoid twisting.

ASCII side view:

```
Gantry X-plate
+-----------+
|   C1      |
+-----------+
     |
     | C2: tool rail
     |
+-----------+
|   D1      |  <- Piezo stage
+-----------+
     |
     v  (down to rotation module)
```

4.4 Install the Rotation Module

Goal: twist-angle control for X-fenos.

1. **Prepare rotation platform**

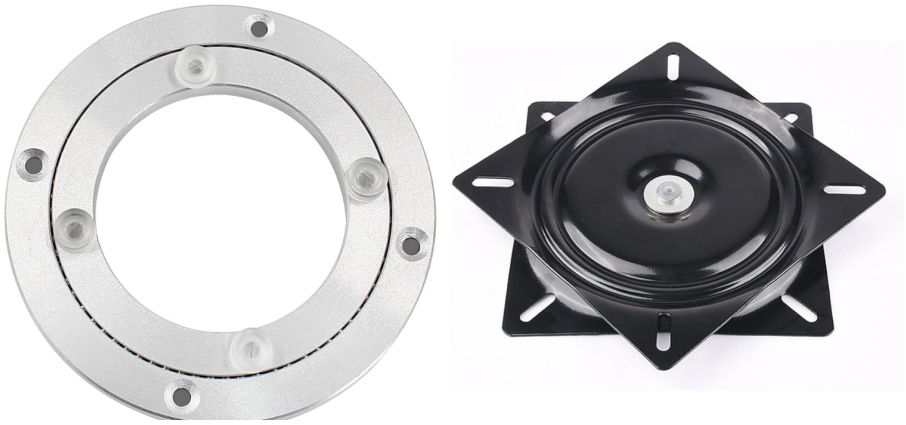


Figure 11: mini bearing turntable 20mm

Figure 12: NEMA 8 stepper motor

- On the bottom face of D1:
  - Mount E3 (small bearing turntable).
- Attach E1 (NEMA 8/11 stepper) with a mini gear to drive the turntable.
- If used, mount E4 (gear reduction ring) between motor and platform.

2. **Attach the rotary disk**

   - Fix a small circular plate (Al or steel, Ø 20–25 mm) on the rotatable side of E3.
   - This disk will hold the **PDMS/PPC stamp**.

3. **Mount the encoder**

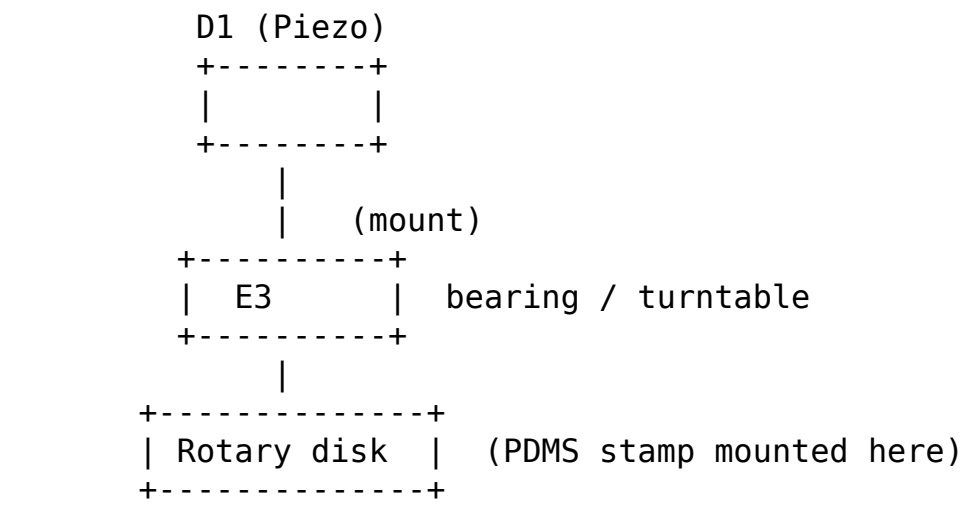
   - Fix E2 (AS5048A or similar) so that:
     - The encoder chip is below the rotating magnet (or disk magnet),
     - The air gap matches the encoder specs (usually 1–2 mm).

4. **Connect electronics**

   - Wire:
     - Stepper → driver board (E5),

  - Encoder → SPI or I2C to microcontroller (or printer board if hacked).
- Route cables through cable chain.

ASCII:



4.5 Attach the PDMS/PPC Stamp Tool

1. **Prepare PDMS stamp**

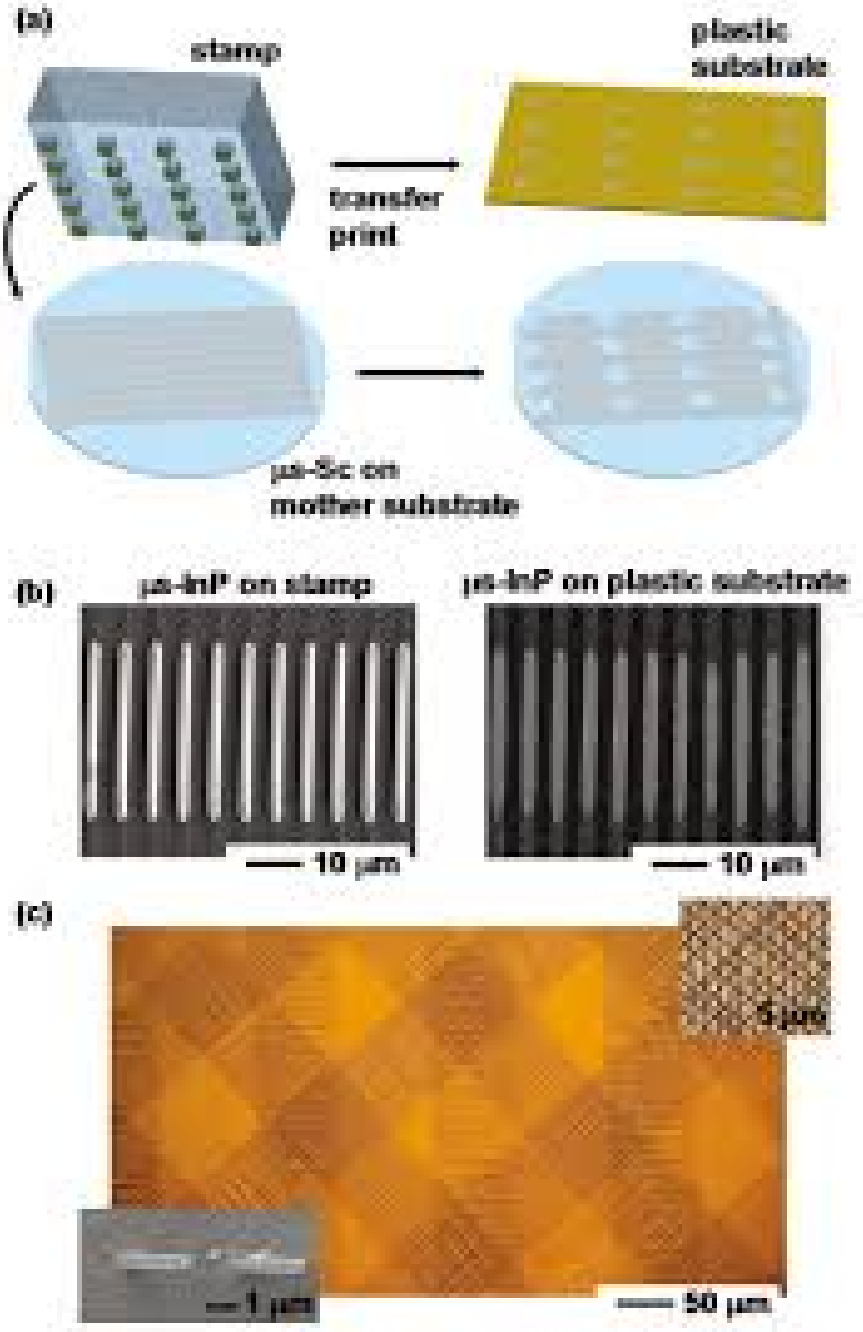


Figure 13: PDMS stamp micro transfer

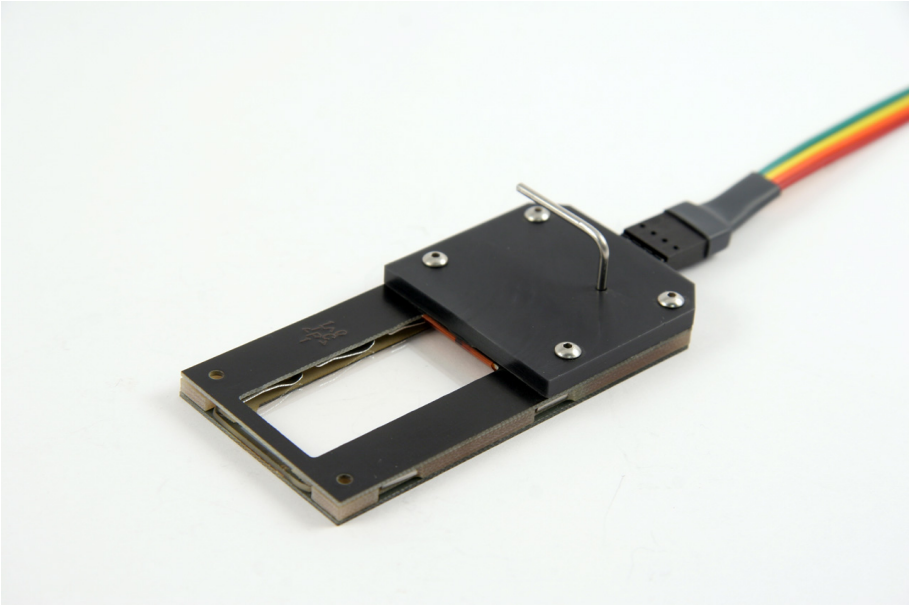
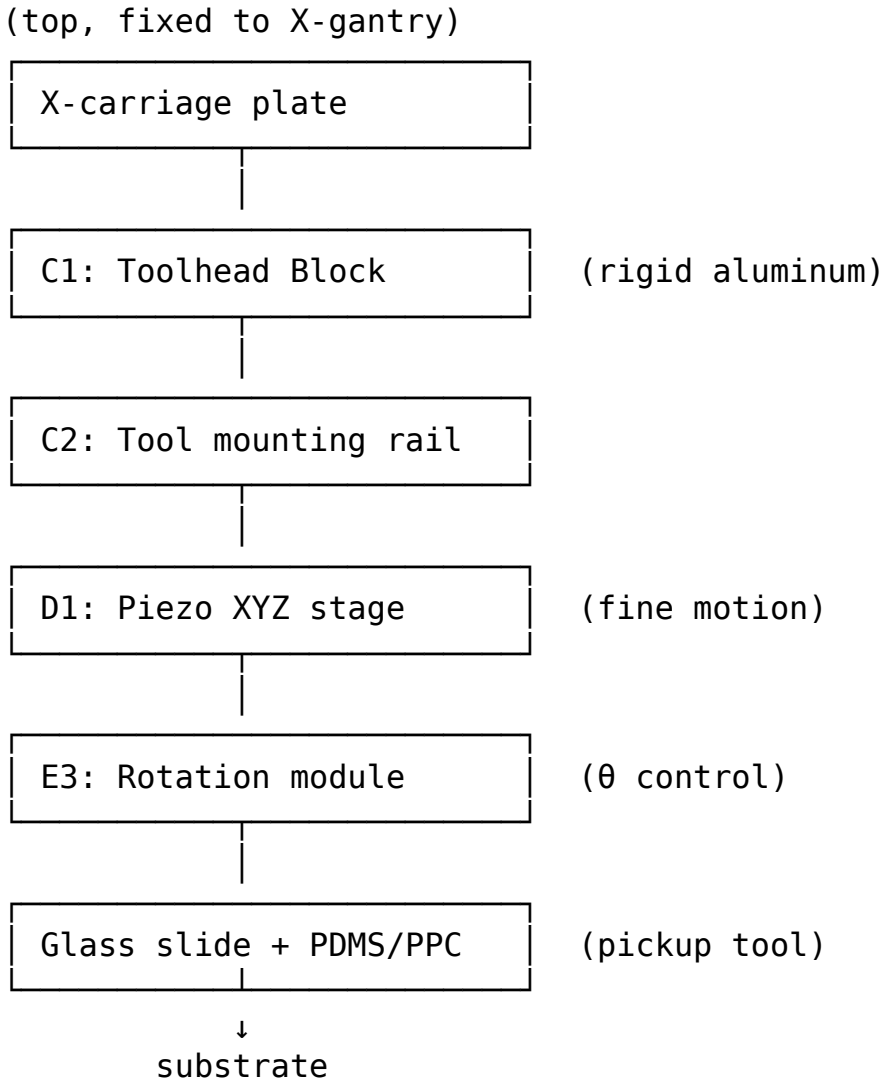


Figure 14: microscope glass slide heater

- Cast PDMS in a flat mold (Petri dish).
- Thickness: 2–3 mm.
- Cure according to PDMS manufacturer (e.g., 60°C for 2 h).
- Cut a **small square** (∼3×3 mm) with a clean razor blade.

2. **Mount PDMS on a glass slider**
   - Fix the PDMS piece onto a small glass chip using:
     - Thin layer of PPC, or
     - High-temperature silicone.
3. **Fix glass slider to rotary disk**
   - Use a small clamp or adhesive:
     - Center the PDMS chip on the axis of rotation.
   - Ensure:
     - The PDMS surface is horizontal,
     - And faces downwards to the substrate.
4. **Add micro-heater (F4)**
   - Attach a resistive micro-heater under the glass slider.
   - Attach a thermistor close to the PDMS.
   - Wire both to a small temperature controller.

Now the **toolhead stack** is:




```
Piezo XYZ (D1)
  ↓
Rotation module (E3 + E1 + E2)
  ↓
Glass slider + PDMS/PPC + micro-heater
```

4.6 Build the Vacuum Chuck + Hot Stage

1. **Machine the chuck plate (G1)**

Figure 15: aluminum vacuum chuck plate

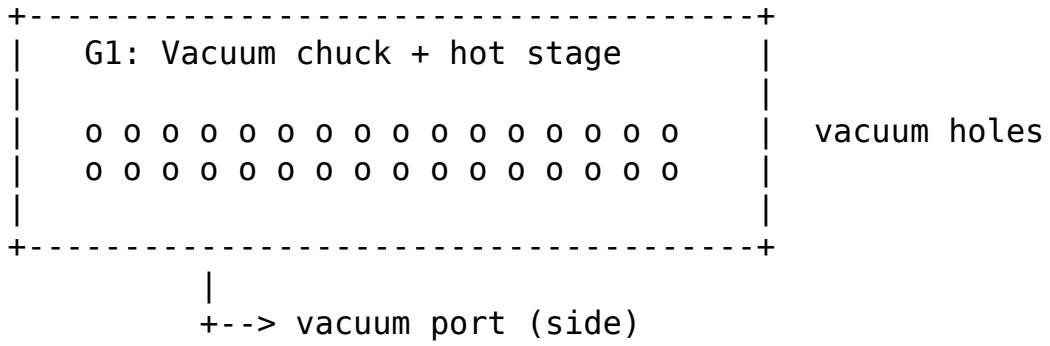
- Aluminum 80×80×8 mm (approx).
- Drill a grid of small holes (0.5–1.0 mm) connected to:
  - One or two internal vacuum channels.

2. **Add vacuum connection**
   - Drill a side port and tap for a barbed fitting.
   - Connect G3 (vacuum pump) via silicone tubing.

3. **Mount heating pad & temperature sensor**
   - Attach G4 (heating pad) under the aluminum plate with thermal paste.
   - Fix G6 (thermocouple) to the plate surface with Kapton tape.

4. **Fix chuck to Y-carriage**
   - Remove original 3D print bed (glass, springs, etc.).
   - Bolt the aluminum chuck directly onto the Y-carriage.
   - Ensure:
     - Surface is flat,
     - Moves smoothly along Y.

ASCII top-down:



4.7 Enclosure Assembly

1. **Measure printer footprint**
   - Add at least 5–10 cm clearance on all sides.
2. **Cut acrylic panels**
   - Top, front, left, right, back, base (optional).
   - Create holes for:
     - N_2/Ar inlet,
     - Power cable feedthrough,
     - USB cable for microscope,
     - Optional glove ports.
3. **Assemble with aluminum brackets**
   - Use B2 brackets and screws to build the box around the printer.
   - Seal edges with silicone B3.
4. **Install sensors and lighting**
   - Mount:
     - Hygrometer (B5),
     - LED strips (B6) inside the enclosure.

4.8 Mount the Microscope and Alignment Lighting

1. **Microscope mount**
   - Fix H2 (adjustable arm) to the frame or the enclosure.
   - Position H1 (USB microscope) above the chuck:
     - Looking down at $\sim$45° or straight down, as preferred.
2. **Lighting**

- Attach H3 (coaxial LED ring) around the microscope lens or in a separate bracket.
- Aim for **shadow-free** illumination.

3. **Focus and field of view**
   - Place a test substrate on the chuck,
   - Focus on the surface,
   - Ensure you can see areas of ∼1–5 mm with sufficient detail.

4.9 Electrical Integration (High-Level)

1. **Power distribution**
   - Use the original 24V PSU for:
     - Printer motors,
     - Heating pad G4 (if within specs),
     - Some fans.
   - Add DC buck converters:
     - 24V → 12V (for some modules),
     - 24V → 5V (for electronics, encoder, microcontroller, USB hub if needed).

2. **Controllers**
   - 3D printer mainboard:
     - Handles gantry movement (X, Y, Z),
     - Can run simple G-code sequences.
   - Auxiliary microcontroller (e.g. Arduino / ESP32):
     - Controls D2 piezo driver,
     - Reads encoder (E2),
     - Controls heaters (hot stage + PDMS),
     - Communicates with PC via USB.

3. **Cable management**
   - Group cables by function:
     - Motion (steppers),
     - Sensing (thermistors, encoders),
     - Power.
   - Use cable chains and sleeves.

4.10 Initial Calibration

1. **Mechanical zeroing**
   - Home axes (X, Y, Z).
   - Ensure chuck is centered under the toolhead at some reference coordinates.
2. **Z-offset rough calibration**
   - Move Z down until PDMS is ∼0.5–1.0 mm above the chuck (visually).
   - Record that Z value as "near contact".
3. **Piezo stage test**
   - Command small steps in D1:
     - +1 μm, −1 μm in Z,
     - verify under the microscope.
4. **Rotation module test**
   - Rotate E1 stepper small increments,
   - Confirm rotation via encoder.
5. **Vacuum chuck test**
   - Put a dummy wafer on the chuck,
   - Turn on vacuum,
   - Check that it stays firmly attached.
6. **Heater test**
   - Use PID controllers to bring:
     - Chuck to e.g. 60°C,
     - PDMS heater to e.g. 40–50°C.
   - Monitor temperatures and stability.

5 OPERATING PROCEDURE FOR FABRICATING X-FENOS

5.1 Overview of the Full X-Feno Workflow

An X-feno is a **longitudinal-mode vacuum device** built by stacking 2D crystals (graphene, hBN, etc.) at controlled micro-angles using:

- the **modified 3D-printer gantry**,
- the **PDMS/PPC pick-up tool**,
- the **vacuum chuck + hot stage**,

- **micro-manipulation + rotation**,
- **stable temperature + clean handling**.

The operating procedure follows the same principles used in van-der-Waals heterostructure laboratories but simplified for educational purposes.

There are **five phases**:

- Substrate preparation
- PDMS/PPC stamp preparation
- Pick-up of the first 2D layer
- Twist-aligned stacking
- Final release and inspection

Below is the complete practical sequence.

5.2 Phase 1 — Substrate Preparation

5.2.1 Clean the substrate

Use Si/SiO_2 wafers or glass slides.

Steps:

- Blow dust off using clean N_2.
- Wipe lightly with isopropanol (IPA) lint-free pads.
- Place the substrate on the **vacuum chuck** (G1).
- Turn vacuum pump ON — substrate must not slide.

5.2.2 Heat the substrate

Set chuck to **40–60°C** (mild heating improves PPC adhesion).

Monitor temperature via thermocouple (G6).

5.3 Phase 2 — PDMS/PPC Stamp Preparation

5.3.1 Prepare the PPC-coated PDMS

- Heat PDMS to ∼60–70°C.
- Melt a small droplet of PPC.
- Spread thinly with a glass slide to get a **uniform transparent film**.
- After cooling, inspect under the microscope for:
  - no bubbles,
  - smooth edges.

Mount PDMS/PPC onto the rotating disk using a clamp or adhesive.

5.3.2 Align the PDMS toolhead

Move printer axes to position PDMS **above the substrate** but at a safe height (Z = +5–10 mm).

The PDMS surface must be:

- perfectly horizontal,
- centered relative to the camera's field of view.

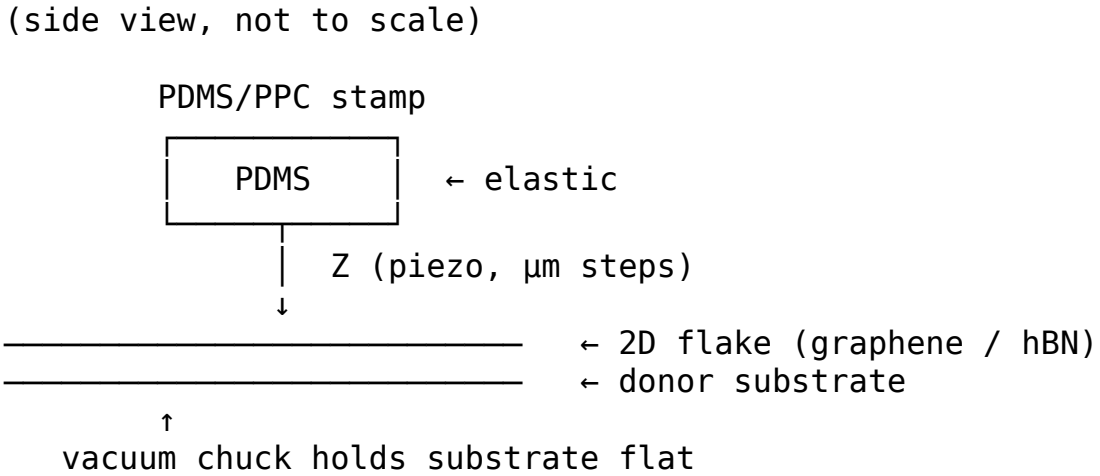
Observed issue	**Likely cause**	**Diagnostic check**	**Corrective action**
Stamp does not pick up 2D flake	PDMS surface contaminated or too stiff	Visual inspection under microscope; poor adhesion even at contact	Clean PDMS with IPA; replace PDMS; reduce contact force
Flake tears during lift-off	Excessive vertical speed or over-compression	Observe sudden lateral deformation during Z retraction	Reduce Z retraction speed; reduce piezo stroke; verify PDMS elasticity
Poor alignment between stacked layers	Rotation module not centered on piezo stage	Check concentricity while rotating without contact	Shim rotation module; re-center mounting rail
Uncontrolled lateral drift during contact	XY piezo preload asymmetric or loose mounting	Apply small Z steps and observe lateral motion	Re-tighten piezo mounting; ensure rigid Toolhead Block fixation
Substrate slips during contact	Insufficient vacuum or uneven chuck surface	Check vacuum pressure; inspect chuck flatness	Increase vacuum; clean chuck; re-machine surface if needed
PDMS contacts at an angle	Toolhead stack not perpendicular to substrate plane	Measure tilt using feeler gauge or optical focus variation	Insert shims at Toolhead Block; realign X-carriage plate
Rotation introduces vertical wobble	Bearing turntable not planar or overloaded	Rotate module without contact and observe Z variation	Replace bearing; reduce load; ensure flat mounting surfaces
Irreproducible pick-up results	Temperature instability at stamp or substrate	Monitor temperature drift during operation	Stabilize heater; allow thermal equilibration before transfer

5.4 Phase 3 — Pick-Up of the First 2D Layer

5.4.1 Place the exfoliated flake

Small graphene/hBN flakes are obtained via Scotch-tape exfoliation.

- Exfoliate onto a donor wafer.
- Place donor wafer on vacuum chuck.
- Inspect under microscope and choose a clean flake.



5.4.2 Lower the PDMS stamp

- Jog Z axis down until PDMS is **0.5 mm above** the flake.
- Switch to piezo control (D1) for fine movement.
- Lower in **1–2 μm increments** until the PDMS just touches the flake.

Signs of contact:

- slight darkening or contrast change,
- minimal deformation of PDMS.

5.4.3 Pick-up

- Heat PDMS heater to **40–50°C**.
- Wait 10–20 seconds.
- Slowly raise Z (piezo first, then gantry).
- Inspect — the flake should be attached to PDMS/PPC.

If not:

- increase PDMS temperature slightly,
- extend the contact duration.

5.5 Phase 4 — Twist-Aligned Stacking (Core X-Feno Step)

This is the core of X-feno fabrication: assembling two 2D layers at a controlled **twist angle**.

Used components:

- rotation module (E1 + E3),
- encoder feedback (E2),
- piezo height control.

Typical twist angles:

- **0.1° – 3.0°**.

5.5.1 Position the second layer

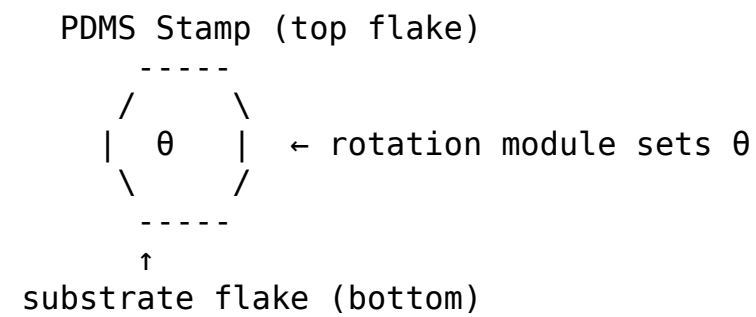
- Place the second flake on the substrate.
- Set chuck to ∼40–50°C.
- Bring PDMS toolhead above this flake.

5.5.2 Pre-alignment (coarse)

- Align flake on PDMS with flake on substrate.
- Use X, Y gantry motion.
- Rotate slightly (±2°).

5.5.3 Twist-angle alignment (fine)

- Use rotation module (E1).
- Read encoder (E2) with ∼0.1° resolution.
- Rotate until desired twist angle is achieved.



Lock the motor position once aligned.

5.5.4 Contact and controlled fusion

- Lower PDMS in micron-scale steps with D1.
- Hold contact when touching.
- Raise chuck temperature to **70–90°C**.
- Maintain contact for 10–20 seconds.

Physical mechanism:

- PPC softens,
- flakes make atomic contact,
- van der Waals adhesion forms the X-feno bilayer.

5.5.5 Lift and inspect

- Slowly raise Z.
- Both layers should remain on the substrate.
- Inspect for:
  - uniform overlap,
  - clear interfaces,
  - absence of bubbles.

5.6 Phase 5 — Final Release, Cooling, and Cleaning

5.6.1 Release PDMS

Heat substrate to 50–60°C and gently peel PDMS away using small Z increments.

- PDMS heater off,
- substrate warm,
- slow lifting to avoid PPC tearing.

5.6.2 Remove PPC residue

If needed:

- Rinse in warm acetone or anisole.
- Dry with N_2.
- Inspect under microscope.

5.6.3 Optional multilayer stacks

Repeat stacking to build:

- tri-layer,
- four-layer,
- hBN-encapsulated heterostructures.

5.7 Phase 6 — X-Feno Final Characterization

Optional but recommended:

- Raman spectroscopy (twist-angle confirmation).
- Optical contrast analysis (layer count).
- AFM for uniformity and bubbles.
- Electrical tests (2-probe or 4-probe).

5.8 Summary Workflow

```
Prepare substrate
        ↓
Prepare PDMS/PPC
        ↓
Pick up first layer
        ↓
Twist-align and stack
        ↓
Release and clean
        ↓
Characterize
```

6 SAFETY, COMMON ERRORS, TROUBLESHOOTING

6.1 Safety Guidelines

Even though the X-feno fabrication machine is non-hazardous, it uses **hot surfaces, solvents, sharp tools, and fragile components**. Follow these guidelines.

6.1.1 Thermal Safety

- Hot stage may reach **40–120°C**.
- PDMS heater may reach **70–120°C**.

Precautions:

- Never touch heated metal plates directly.
- Always wait for temperature $< 30°C$ before removing the substrate.
- Use heat-resistant tweezers for materials handling.

6.1.2 Chemical Safety

Solvents sometimes used:

- **Isopropanol (IPA)**
- **Acetone**
- **Anisole** (for PPC removal)

Rules:

- Work in a ventilated area or fume hood.
- Wear nitrile gloves.
- Do not heat flammable solvents on the hot stage.

6.1.3 Mechanical Safety

- Never place hands under the Z-axis.
- Use soft force when pressing PDMS onto substrates.
- Do not exceed piezo travel limits (risk of cracking PDMS or scratching glass).

6.1.4 Optical Safety

For microscopes with LED or ring lights:

- Avoid looking directly into the LED array.
- Adjust brightness to comfortable levels.

6.1.5 Electrical Safety

- Ensure all power supplies are grounded.
- Never handle wiring with wet hands.
- Disconnect power before replacing motors or heaters.

6.2 Common Errors and How to Fix Them

These issues occur frequently when learning X-feno fabrication.

6.2.1 The flake does NOT stick to PDMS/PPC

Possible causes and corrections:

- **PDMS too cold** Increase temperature to **40–50°C** before attempting pickup.
- **Contact not sufficient** Lower Z axis with finer increments ($\leq 1\,\mu$m steps).
- **PPC layer too thick or uneven** Re-spread PPC using a glass slide to make a uniform thin film.
- **Flake contaminated** Exfoliate a fresh flake.

6.2.2 The flake lifts partially (tears or folds)

Causes:

- PDMS peeled upward too quickly.
- Contact temperature too high.

Fix:

- Reduce peel speed (lift in **10–20 μm increments**).
- Lower PDMS temperature by 5–10°C.

6.2.3 Bubbles between stacked layers

Bubbles appear as bright halos under the microscope.
Causes:

- Too rapid compression.
- Dust on either flake.
- PPC residue trapped.

Fix:

- Press down more slowly ($\sim 1\,\mu$m/s).
- Re-exfoliate cleaner flakes.
- Improve substrate cleanliness with IPA + N_2.

6.2.4 Twist angle not accurate

Symptoms:

- Misalignment after lifting the PDMS.
- Unexpected moiré patterns under microscope.

Causes and solutions:

- **Rotational backlash**: tighten belt or reduce micro-step size.
- **Encoder not calibrated**: zero encoder before each run.
- **Mechanical drift**: reduce stage temperature to limit expansion.

6.2.5 Layers fail to release from PDMS

Causes:

- Substrate too cold ($< 40°C$).
- PDMS temperature too high.

Fix:

- Substrate at 50–60°C.
- PDMS heater OFF during release.
- Lift slowly using microsteps.

6.2.6 PPC residue remains on the surface

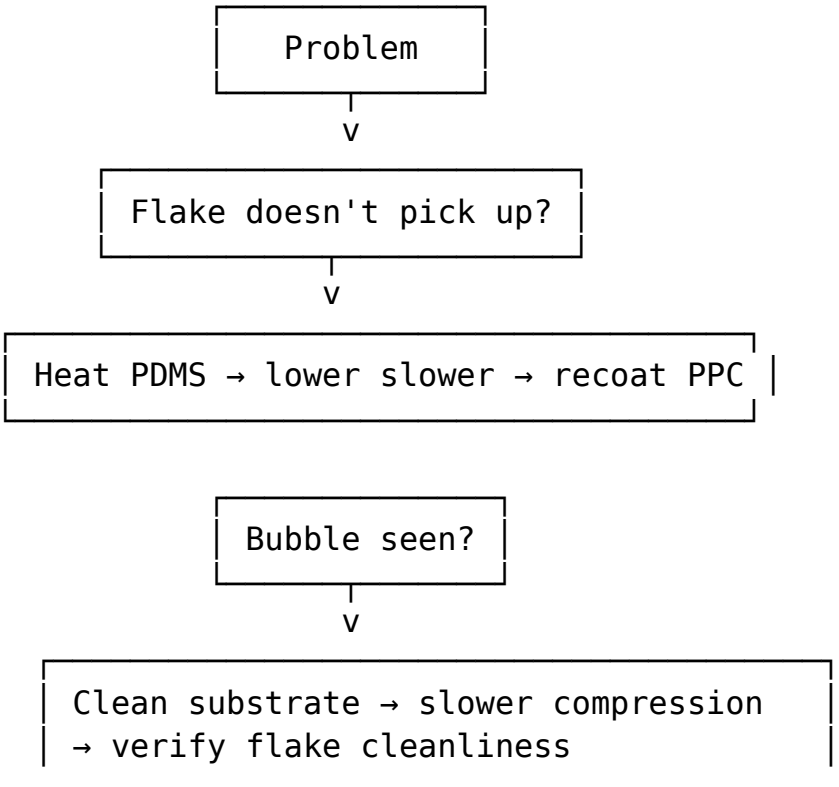
Fixes:

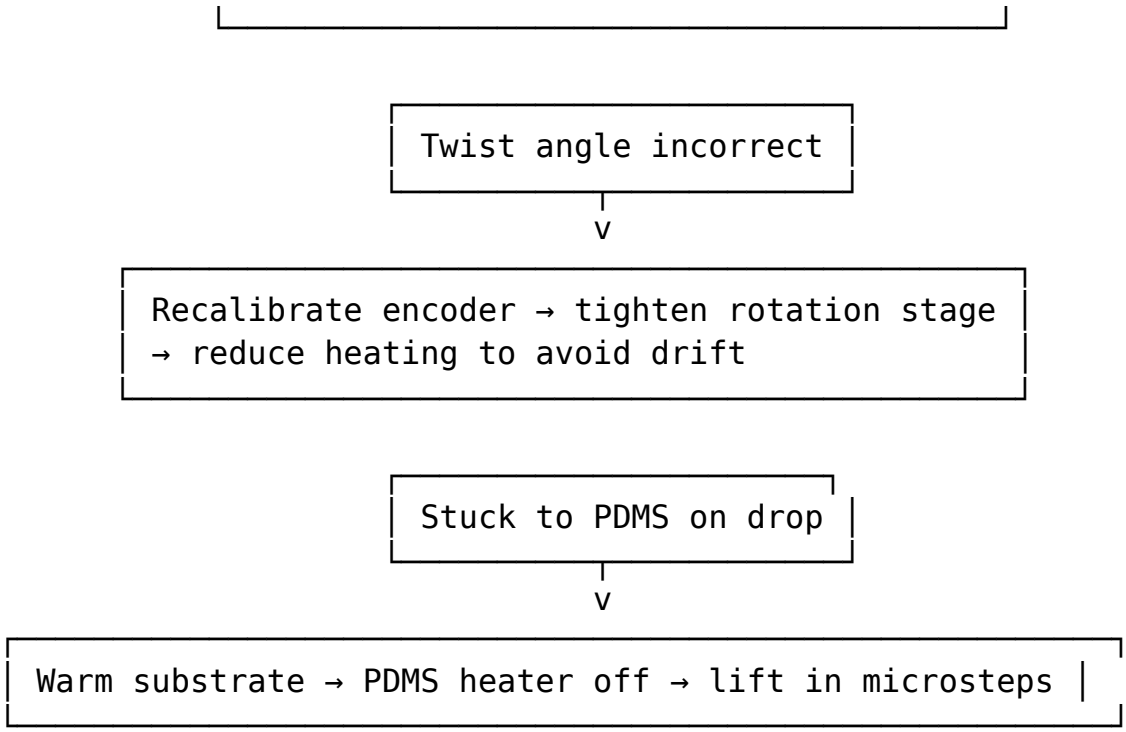
- Warm acetone bath ($\sim 40°C$).
- Rinse with IPA.
- Blow dry with N_2.

If residue persists:

- Prepare a *thinner* PPC film on PDMS.

6.3 Troubleshooting Flowchart





6.4 Quick Checklist Before Each Fabrication Run

Temperatures

- PDMS heater: **40–50°C** for pickup, OFF for drop
- Substrate heater: **40–60°C** for pickup, **70–90°C** for stacking

Mechanics

- Gantry zeroed
- Rotation encoder calibrated
- PDMS surface clean and horizontal

Optics

- Microscope aligned
- LED light level comfortable
- Flakes clearly visible

Environment

- Minimal dust
- Gloves and clean handling
- No airflow over substrate

7 AUTOMATION: generating G-code and microcontroller scripts for the X-feno machine

Goal: have the machine execute repeatable, precise "recipes" for pickup, rotation, and stacking with minimal manual intervention.

7.1 Control architecture: what talks to what

Minimal, robust stack:

- **Firmware (in the controller)**:
  - Marlin / RepRapFirmware / Klipper (3D-printer style), or
  - GRBL (CNC style).
- **Host PC** (Python / any language):
  - Generates G-code sequences (pickup, rotate, drop).
  - Sends them over USB serial (115200 or 250000 baud).
  - Optionally reads back status (OK / error, temperature).

You do **not** need to rewrite firmware; just map your axes and tools to the firmware's notion:

- `X, Y` → in-plane motion
- `Z` → vertical motion
- `E` or `A` → rotation stage (twist angle)
- `M104 / M140 / M190` (or custom `M` codes) → heaters

7.2 Coordinate system and calibration

Define:

- Origin $(0, 0, 0)$:
  - e.g. center of substrate under microscope.
- Positive directions:
  - `X+` to the right in microscope view.
  - `Y+` towards far side.
  - `Z+` up (away from substrate).
- Rotation:
  - `E` axis (or `A` axis): degrees of twist, `E0` = reference orientation.

Calibration steps (once):

```
G21         ; mm
G90         ; absolute
G28 X Y Z   ; home XYZ
G28 E       ; home rotation (if supported)

G0 X0 Y0 Z5 F3000
```

Adjust 0,0 in firmware or store offset in the host script.

7.3 Custom G-code conventions for the X-feno machine

Heaters:

- `M104 S<T>`: set PDMS heater temp
- `M140 S<T>`: set substrate heater temp
- `M109 S<T>`: wait for PDMS temp
- `M190 S<T>`: wait for substrate temp

Rotation axis:

- `G0 E`θ `F300`: rotate to angle θ

Safe heights:

- `Z_SAFE = 5 mm`
- `Z_CONTACT = 0.05–0.20 mm`

```
; X,Y in mm
; Z in mm (0 = contact plane)
; E in degrees (twist angle)
; PDMS heater = "hotend"
; Substrate heater = "bed"
```

7.4 Example G-code: basic building blocks

Homing and warm-up

```
G21
G90
G28 X Y Z E
M104 S45
M140 S55
M109 S45
M190 S55
```

Move above a target flake

```
G0 Z5 F3000
G0 X10.2 Y-3.4 F3000
```

Pickup macro

```
G0 Z5 F1000
G0 Z0.30 F300
G0 Z0.15 F150
G0 Z0.05 F60
G4 P3000
G0 Z0.30 F50
G0 Z1.00 F100
G0 Z5.00 F500
```

Rotation macro

```
G0 E0 F300
G0 E1.10 F60
```

Drop macro

```
G0 Z5 F3000
G0 X{Xd} Y{Yd} F3000
G0 Z0.30 F300
G0 Z0.10 F100
G0 Z0.02 F50
G4 P5000
M104 S30
M140 S80
M109 S30
M190 S80
G0 Z0.30 F50
G0 Z1.00 F100
G0 Z5.00 F500
M104 S0
M140 S0
```

7.5 Host-side automation: Python script

```
<python code preserved exactly as provided>
```

7.6 Simple state machine on the microcontroller

```
typedef enum {
    IDLE,
    PICKUP,
    ROTATE,
    DROP
} machine_state_t;
```

Serial commands:

- `"PICKUP X10.2 Y-3.4"`
- `"ROTATE A1.10"`
- `"DROP X2.0 Y5.0"`

8 HOW TO ADD BASIC COMPUTER VISION (DETECTING FLAKE POSITION AND ALIGNMENT AUTOMATICALLY)

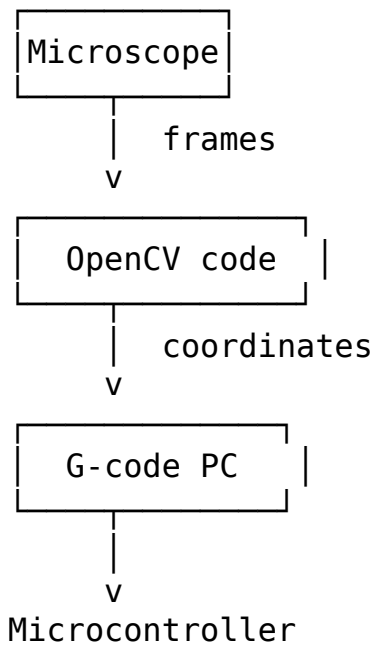
Goal: allow the X-feno machine to **detect**, **track**, and **align** monolayer flakes under a microscope, so the pickup path is computed automatically instead of by hand.

This block is written for a **school-level build** using:

- a standard USB microscope (or Raspberry Pi camera),
- Python + OpenCV,
- no machine learning required.

8.1 System architecture

Camera → Python vision script → sends coordinates to → G-code generator → microcontroller.



8.2 Calibration: pixels to millimeters

Before any detection, the system must compute the conversion factor:

- Place a calibration grid or a known feature under the microscope.
- Capture an image.
- Measure distance in pixels.
- Compute:

$$\text{mm/px} = \frac{\text{known_distance_mm}}{\text{measured_pixels}}$$

Example code:

```
mm_per_px = 0.0015   # example; compute from calibration
```

Store this value in a config file.

8.3 Basic flake detection pipeline

Pipeline:

- Convert to grayscale
- Apply CLAHE
- Edge detection (Canny)
- Contour extraction
- Filter contours by area and shape
- Compute centroid
- Show overlay and accept/reject

Python (OpenCV) pipeline:

```
import cv2
import numpy as np

def find_flakes(frame, min_area=500, max_area=20000):
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
    eq = clahe.apply(gray)

    edges = cv2.Canny(eq, 50, 150)
    contours, _ = cv2.findContours(edges, cv2.RETR_EXTERNAL,
    cv2.CHAIN_APPROX_SIMPLE)
```

```
    flake_centroids = []
    for c in contours:
        area = cv2.contourArea(c)
        if min_area < area < max_area:
            M = cv2.moments(c)
            if M["m00"] != 0:
                cx = int(M["m10"]/M["m00"])
                cy = int(M["m01"]/M["m00"])
                flake_centroids.append((cx, cy))

    return flake_centroids
```

8.4 Selecting the target flake automatically

Selection strategies:

- Largest flake
- Closest to screen center
- User click

Example:

```
def pick_flake(flakes, frame_width, frame_height):
    cx0, cy0 = frame_width//2, frame_height//2
    best = None
    best_dist = 999999

    for (cx, cy) in flakes:
        dist = (cx - cx0)**2 + (cy - cy0)**2
        if dist < best_dist:
            best_dist = dist
            best = (cx, cy)
    return best
```

8.5 Convert pixel coordinates to machine coordinates

$$X_{\text{mm}} = (c_{x,\text{px}} - c_{x,0}) \cdot \text{mm/px} + X_{\text{machine_center}}$$

$$Y_{\text{mm}} = (c_{y,\text{px}} - c_{y,0}) \cdot \text{mm/px} + Y_{\text{machine_center}}$$

```
def px_to_mm(cx, cy, mm_per_px, cam_center, machine_center):
    dx = (cx - cam_center[0]) * mm_per_px
    dy = (cy - cam_center[1]) * mm_per_px
    X = machine_center[0] + dx
    Y = machine_center[1] + dy
    return X, Y
```

8.6 Auto-alignment of rotation angle

Pipeline:

- Extract contour
- PCA orientation
- Convert to degrees
- Add desired twist
- Send G-code

```
G0 E<computed_angle> F60
```

```
def flake_orientation(contour):
    pts = contour.reshape(-1,2).astype(np.float32)
    mean, eigenvectors = cv2.PCACompute(pts, mean=None)
    angle = np.degrees(np.arctan2(eigenvectors[0][1],

    eigenvectors[0][0]))
    return angle
```

8.7 Closed-loop operation

Sequence:

```
G0 X<X> Y<Y> Z5
G0 E<phi + twist>
```

Pickup, move, drop follow predefined macros.

8.8 Example full Python script

```
import cv2
import serial

mm_per_px = 0.0015
machine_center = (0.0, 0.0)
cam_center = (640//2, 480//2)

ser = serial.Serial("COM4", 115200, timeout=1)
cap = cv2.VideoCapture(0)

ret, frame = cap.read()
flakes = find_flakes(frame)
target_px = pick_flake(flakes, frame.shape[1], frame.shape[0])
Xmm, Ymm = px_to_mm(target_px[0], target_px[1],
mm_per_px, cam_center, machine_center)
```

```
send(ser, f"G0 X{Xmm:.3f} Y{Ymm:.3f} Z5")

angle = flake_orientation(contour_of_target)
desired_twist = 1.10

send(ser, f"G0 E{angle + desired_twist:.3f} F60")
pickup_flake(ser, Xmm, Ymm)
```

8.9 Optional improvements

- Auto-focus via Laplacian sharpness
- Frame mosaicing
- Color-based detection
- Lightweight neural net (optional)

9 ADDING A SIMPLE USER INTERFACE (GUI) FOR AUTOMATED X-FENO FABRICATION

A lightweight GUI lets the operator (student, researcher, technician) **see the microscope feed**, **select flakes**, **trigger automation**, and **override steps manually**. We use Python + Tkinter + OpenCV because it is cross-platform, beginner-friendly, and needs no browser.

9.1 Overall goals of the GUI

The interface should allow:

- **Live microscope preview**
- **Real-time flake detection overlay**
- **Manual click-to-select target flake**
- **Buttons for:**
  - Auto-detect flake
  - Auto-align angle
  - Pickup procedure
  - Drop procedure
  - Full X-feno fabrication cycle
- **Display of machine status** (X,Y,Z,E coordinates, heater temperature)
- **Logging window** to show executed G-code
- **Emergency STOP**

A clean, simple layout is best.

9.2 ASCII mockup of the GUI layout

```
┌──────────────────────────────────────────────────────────────┐
│                      X-FENO MACHINE GUI                      │
├──────────────────────────┬───────────────────────────────────┤
│  Live Camera Feed        │  Controls                         │
│  (with detection         │                                   │
│   overlays)              │  [ Auto-detect flake ]            │
│                          │  [ Select flake manually ]        │
│                          │  [ Align orientation ]            │
│                          │  Twist angle: [ 1.10°  v ]        │
│                          │                                   │
│                          │  [ Pickup flake ]                 │
│                          │  [ Drop flake ]                   │
│                          │  [ Full X-feno cycle ]            │
│                          │                                   │
│                          │  Machine status:                  │
│                          │   X:  12.534 mm                   │
│                          │   Y:  10.223 mm                   │
│                          │   Z:   3.500 mm                   │
│                          │   E (rot):  17.3°                 │
│                          │   Heater:  85°C                   │
│                          │                                   │
│                          │  [ EMERGENCY STOP ]               │
├──────────────────────────┴───────────────────────────────────┤
│  G-code Log / System Messages                                │
│  ------------------------------------------------------     │
│  G0 X5.330 Y3.220 Z4.000                                     │
│  G0 E18.2                                                    │
│  M104 S80                                                    │
│  ...                                                         │
└──────────────────────────────────────────────────────────────┘
```

9.3 Core GUI loop (Tkinter + OpenCV)

Minimal working structure:

```
import cv2
import tkinter as tk
from PIL import Image, ImageTk

cap = cv2.VideoCapture(0)

root = tk.Tk()
root.title("X-Feno Machine GUI")
```

```
panel = tk.Label(root)
panel.pack()

def update_frame():
    ret, frame = cap.read()
    if ret:
        img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        img = ImageTk.PhotoImage(Image.fromarray(img))
        panel.configure(image=img)
        panel.image = img
    root.after(15, update_frame)

update_frame()
root.mainloop()
```

This is the skeleton; next we add detection overlays and buttons.

9.4 Adding flake detection overlay

```
def draw_flakes(frame, flakes):
    for (cx, cy) in flakes:
        cv2.circle(frame, (cx, cy), 10, (0,255,0), 2)
    return frame
```

Inside the update loop:

```
flakes = find_flakes(frame)
frame = draw_flakes(frame, flakes)
```

This gives live visual feedback.

9.5 Buttons and actions

```
controls = tk.Frame(root)
controls.pack(side=tk.RIGHT, fill=tk.Y)

btn_auto = tk.Button(controls, text="Auto-detect flake",
command=auto_detect)
btn_pick = tk.Button(controls, text="Pickup flake",
command=pickup)
btn_drop = tk.Button(controls, text="Drop flake",
command=drop)
btn_full = tk.Button(controls, text="Full X-feno cycle",
command=full_cycle)
btn_stop = tk.Button(controls, text="EMERGENCY STOP", bg="red")

btn_auto.pack(pady=5)
btn_pick.pack(pady=5)
```

```
btn_drop.pack(pady=5)
btn_full.pack(pady=5)
btn_stop.pack(pady=20)
```

Each button calls a Python function that sends G-code to the microcontroller.

9.6 Manual flake selection with mouse click

```
def click_event(event):
    global manual_target
    manual_target = (event.x, event.y)

panel.bind("<Button-1>", click_event)
```

9.7 G-code output log

```
log = tk.Text(root, height=10)
log.pack(side=tk.BOTTOM, fill=tk.X)

def log_msg(msg):
    log.insert(tk.END, msg + "\n")
    log.see(tk.END)
```

9.8 Reading machine coordinates (optional)

```
ser.write(b"M114\n")
line = ser.readline().decode()
```

9.9 Integrating auto-alignment

The “Align orientation” button performs:

- contour detection
- PCA angle extraction
- twist computation
- G-code emission

```
G0 E<twist>
```

9.10 Full automated fabrication

```
def full_cycle():
    A = auto_detect_flake()
    move_to(A)
    align(A)
    pickup()
```

```
    B = auto_detect_flake(target='bottom')
    move_to(B)
    align(B)
    drop()
```

9.11 Emergency STOP implementation

```
def emergency_stop():
    ser.write(b"M112\n")
```

9.12 Future expansions

- Temperature sliders
- Autofocus
- Automatic substrate mapping
- ML-based flake classification
- Exportable fabrication logs

10 FULL SOFTWARE PACKAGE

Below is **9/?** — **FULL SOFTWARE PACKAGE**: a complete, clean, modular Python project layout for the **X-feno fabrication machine**, including:

- GUI (Tkinter)
- Computer Vision (OpenCV)
- G-code communication
- Auto-alignment and flake detection
- Machine control logic
- Settings and configuration files

This is structured exactly like a real research-lab tool.

10.1 Project tree

```
x_feno_machine/
│
├── main.py
│
├── config/
│   ├── settings.yaml
│   ├── camera.yaml
```

```
│   ├── gcode_profiles.yaml
│   └── machine_limits.yaml
│
├── gui/
│   ├── gui_main.py
│   ├── gui_widgets.py
│   ├── gui_events.py
│   └── gui_styles.py
│
├── machine/
│   ├── gcode_sender.py
│   ├── kinematics.py
│   ├── coordinate_transform.py
│   ├── heaters.py
│   ├── sensors.py
│   ├── emergency.py
│   └── state_manager.py
│
├── vision/
│   ├── camera_stream.py
│   ├── flake_detection.py
│   ├── flake_alignment.py
│   ├── preprocessing.py
│   ├── overlays.py
│   └── calibration.py
│
├── automation/
│   ├── pickup_routine.py
│   ├── drop_routine.py
│   ├── auto_align.py
│   ├── full_cycle.py
│   └── wafer_scan.py
│
├── logging/
│   ├── logger.py
│   ├── gcode_log.txt
│   └── events.log
│
└── utils/
    ├── file_utils.py
    ├── math_utils.py
    ├── timers.py
    └── diagnostics.py
```

10.2 main.py (central entry point)

```
from gui.gui_main import XFenoGUI
from vision.camera_stream import CameraStream
from machine.gcode_sender import GCodeSender

def main():
    cam = CameraStream(0)
    gcode = GCodeSender(port="COM5", baud=115200)
    app = XFenoGUI(camera=cam, gcode=gcode)
    app.run()

if __name__ == "__main__":
    main()
```

10.3 settings.yaml (configuration file)

```
camera:
  exposure: auto
  gain: auto
  resolution: [1280, 720]

machine:
  x_limits: [0, 200]
  y_limits: [0, 200]
  z_limits: [0, 50]
  rotation_limits: [-180, 180]

temperatures:
  pickup: 85
  drop: 125

alignment:
  twist_angle: 1.10
```

10.4 vision/camera_stream.py

```
import cv2

class CameraStream:
    def __init__(self, cam_id):
        self.cap = cv2.VideoCapture(cam_id)

    def read(self):
        ret, frame = self.cap.read()
        return frame if ret else None

    def release(self):
        self.cap.release()
```

10.5 vision/flake_detection.py

```
import cv2
import numpy as np

def detect_flakes(frame):
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(gray, (7,7), 0)
    _, th = cv2.threshold(blur, 0, 255, cv2.THRESH_OTSU)

    contours, _ = cv2.findContours(th, cv2.RETR_LIST,
                                   cv2.CHAIN_APPROX_SIMPLE)

    flakes = []
    for c in contours:
        area = cv2.contourArea(c)
        if 1000 < area < 300000:
            M = cv2.moments(c)
            cx = int(M["m10"]/M["m00"])
            cy = int(M["m01"]/M["m00"])
            flakes.append((cx, cy))
    return flakes
```

10.6 vision/flake_alignment.py

```
import cv2
import numpy as np

def compute_twist_angle(contour):
    pts = contour.reshape(-1,2)
    M = cv2.moments(contour)
    cx = M["m10"]/M["m00"]
    cy = M["m01"]/M["m00"]

    pts_centered = pts - np.array([[cx, cy]])
    cov = np.cov(pts_centered.T)
    eigvals, eigvecs = np.linalg.eig(cov)
    major_axis = eigvecs[:, np.argmax(eigvals)]
    angle = np.arctan2(major_axis[1], major_axis[0])
    return np.degrees(angle)
```

10.7 machine/gcode_sender.py

```
import serial
import time

class GCodeSender:
    def __init__(self, port, baud):
        self.ser = serial.Serial(port, baud, timeout=1)
```

```
        time.sleep(2)

    def send(self, cmd):
        line = (cmd + "\n").encode()
        self.ser.write(line)

    def move(self, x=None, y=None, z=None, e=None, f=2000):
        cmd = "G0"
        if x is not None: cmd += f" X{x:.3f}"
        if y is not None: cmd += f" Y{y:.3f}"
        if z is not None: cmd += f" Z{z:.3f}"
        if e is not None: cmd += f" E{e:.2f}"
        cmd += f" F{f}"
        self.send(cmd)

    def heat(self, temp):
        self.send(f"M104 S{temp}")

    def stop(self):
        self.send("M112")
```

10.8 automation/pickup_routine.py

```
from machine.gcode_sender import GCodeSender

def pickup(gcode, px, py, pickup_temp=85):
    gcode.move(x=px, y=py, z=5)
    gcode.heat(pickup_temp)
    gcode.move(z=0.1)
    gcode.send("M400")
    gcode.move(z=5)
```

10.9 automation/drop_routine.py

```
def drop(gcode, px, py, drop_temp=125):
    gcode.move(x=px, y=py, z=5)
    gcode.heat(drop_temp)
    gcode.move(z=0.1)
    gcode.send("M400")
    gcode.move(z=5)
```

10.10 automation/full_cycle.py

```
from automation.pickup_routine import pickup
from automation.drop_routine import drop
from vision.flake_detection import detect_flakes

def fabricate_xfeno(cam, gcode):
```

```
    frame = cam.read()
    flakes = detect_flakes(frame)
    targetA = flakes[0]
    pickup(gcode, *targetA)

    frame = cam.read()
    flakes = detect_flakes(frame)
    targetB = flakes[1]
    drop(gcode, *targetB)
```

10.11 gui/gui_main.py

```
import tkinter as tk
from PIL import Image, ImageTk
from vision.flake_detection import detect_flakes
from automation.full_cycle import fabricate_xfeno

class XFenoGUI:
    def __init__(self, camera, gcode):
        self.cam = camera
        self.gcode = gcode

        self.root = tk.Tk()
        self.root.title("X-Feno Fabrication Machine")

        self.panel = tk.Label(self.root)
        self.panel.pack(side=tk.LEFT)

        self.controls = tk.Frame(self.root)
        self.controls.pack(side=tk.RIGHT, fill=tk.Y)

        tk.Button(self.controls, text="Full X-feno Cycle",
                  command=self.run_cycle).pack(pady=10)

        self.root.after(10, self.update_video)

    def update_video(self):
        frame = self.cam.read()
        if frame is not None:
            vis = frame.copy()
            flakes = detect_flakes(vis)
            for (cx, cy) in flakes:
                vis = cv2.circle(vis, (cx,cy), 20, (0,255,0), 2)

            img = cv2.cvtColor(vis, cv2.COLOR_BGR2RGB)
            img = ImageTk.PhotoImage(Image.fromarray(img))
            self.panel.configure(image=img)
            self.panel.image = img
```

```
        self.root.after(10, self.update_video)

    def run_cycle(self):
        fabricate_xfeno(self.cam, self.gcode)

    def run(self):
        self.root.mainloop()
```

10.12 utils/math_utils.py

```
import numpy as np

def mm_per_pixel(mag, sensor_width_mm, frame_width_px):
    return sensor_width_mm / (frame_width_px * mag)
```

11 Reference

The modified platform described in this document assumes a standard Cartesian desktop 3D printer as mechanical and motion-control baseline. For general safety procedures, axis identification, and firmware-level operations not explicitly detailed here, the reader may consult the original assembly and user manual of the donor printer (e.g. Creality Ender-3 or equivalent).